



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 1 672 543 A2**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
21.06.2006 Bulletin 2006/25

(51) Int Cl.:
G06F 17/30 (2006.01)

(21) Application number: **05111509.5**

(22) Date of filing: **30.11.2005**

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI
SK TR**
Designated Extension States:
AL BA HR MK YU

(72) Inventors:
• **Villaron, Shawn A.**
Redmond, WA 98052 (US)
• **Garg, Sharad K.**
Redmond, WA 98052 (US)

(30) Priority: **20.12.2004 US 18910**

(74) Representative: **Zimmer, Franz-Josef**
Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(71) Applicant: **MICROSOFT CORPORATION**
Redmond, WA 98052 (US)

(54) **File formats, methods, and computer program products for representing presentations**

(57) File formats, methods, and computer program products are provided for representing a presentation in a modular content framework. The modular content framework may include a file format container associated with modular parts. A file format includes logically separate modular parts that are associated with each other by one or more relationships where each modular part is associated with a relationship type. The modular parts include a presentation part representing a start part for the presentation, a slide master part associated with the presentation part, and a slide layout part associated with the slide master part. The modular parts may also include a document properties part containing built-in properties associated with the file format, a thumbnail part containing associated thumbnails, and a slide part containing a slide in the presentation. Each modular part is capable of being interrogated separately, extracted from the presentation, and/or reused in a different presentation.

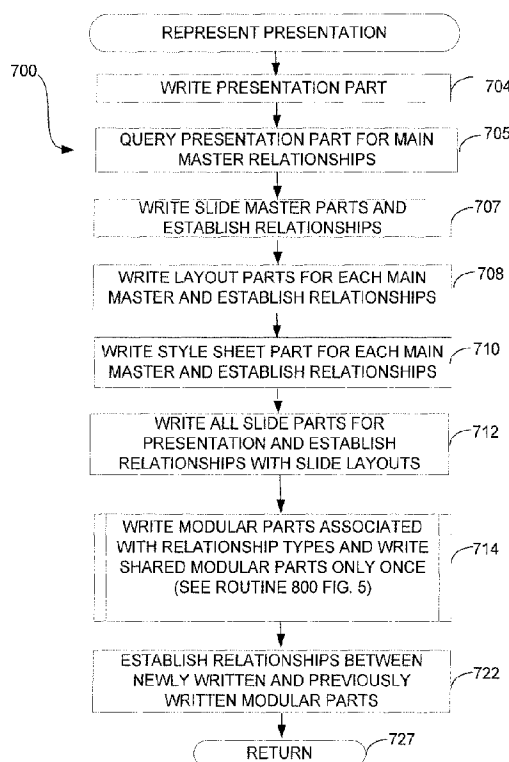


Fig. 4

EP 1 672 543 A2

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application is related to and filed with U.S. Patent Application, Attorney Docket No. 60001.0441US01, entitled "File Formats, Methods, and Computer Program Products For Representing Documents," filed on December 20, 2004; U.S. Patent Application, Attorney Docket No. 60001.0440US01, entitled "Management and Use of Data in a Computer-Generated Document," filed on December 20, 2004; and U.S. Patent Application, Attorney Docket No. 60001.0447US01, entitled "File Formats, Methods, and Computer Program Products For Representing Workbooks," filed on December 20, 2004; all of which are assigned to the same assignee as this application. The aforementioned patent applications are expressly incorporated herein, in their entirety, by reference.

TECHNICAL FIELD

[0002] The present invention generally relates to file formats, and more particularly, is related to methods and formats for representing presentations in a componentized presentation application program.

BACKGROUND

[0003] The information age has facilitated an era of building informative slide presentations utilizing presentation software applications. However, the way in which previous file formats are created and structured to store a collection of graphical surfaces, typically called a presentation, has several drawbacks. For instance, previous presentation file formats are created in the form of a single file containing monolithic data. Because proprietary formats are generally used to create these single files, each company that builds presentation storage develops a different file format. Because the data within these different file formats is monolithic and inaccessible in discrete parts, a series of problems are created.

[0004] One problem is basic document or slide re-use. For instance, it is very difficult to extract one or more slides from a presentation and reuse the extracted slides in a different presentation and retain slide integrity, even in the same application. Comparatively, reusing slides between different applications, for example from POWERPOINT to HARVARD GRAPHICS, and HARVARD GRAPHICS to FREE LANCE GRAPHICS, is worse. Reusing content on a slide is similarly difficult.

[0005] Secondly, because of the single file format, it is practically impossible to lock part of a presentation. Thus, a feature such as multi-user editing, where a number of people perhaps on different platforms, and/or from different locales cooperatively edit a presentation with the help of a locking mechanism, is prohibited. Most of the technology in terms of locking is all done at the file level,

thus if a file is locked by a user, no other users can edit the file. Viewing is possible, but not editing.

[0006] There is also a problem of document interrogation. Finding content within a presentation, for example finding slides for a 2004 sales forecast, can be a daunting task. It is very difficult to find discrete parts within a monolithic file format presentation where semantics of the content can be determined. Even if a tool is built to interrogate one previous file format, the same tool could not be used to interrogate a different previous file format. This problem exists even when an existing binary file format is documented. It is still difficult to implement reader and writer classes that can handle existing binary file formats well. Even if a tool targeted at an application was developed it could not interrogate all document formats. This problem is referred to as the opaqueness of single file formats.

[0007] Still further, due to intermingling of data, the ability to re-brand a presentation, or multiple presentations, is nearly impossible outside of the presentation application. Re-branding a presentation involves taking a slide from presentation A, moving it to presentation B, and making the slide look as though it was authored in the normal authoring context of presentation B. Although, the slide may pick up some colors from the presentation B, the slide will not be displayed as though it was actually authored in presentation B.

[0008] Document surfacing, the ability to take pieces of one file formatted document and drop them into another document, is also a problem. For instance, a spreadsheet table copied from a spreadsheet document into a presentation document is difficult to interrogate in a monolithic style file format.

[0009] Still further, in the case of document previewing, for instance graphically browsing accessible content, it is very difficult to retrieve a high resolution preview of the content exposed through a shell in a browser or in a third party application. Some presentation applications provide thumbnails or previews of a single slide, but none provide high-resolution previews of all of the slides in a presentation deck.

[0010] Accordingly there is an unaddressed need in the industry to address the aforementioned deficiencies and inadequacies.

SUMMARY

[0011] Embodiments of the present invention provide file formats, methods, and computer program products for representing a presentation in a modular content framework implemented within a computing apparatus. Embodiments of the present invention disclose an open file format, such as an extensible markup language (XML) file format, and a way of creating a file format for storing a collection of graphical surfaces, typically called a presentation. A single graphical surface may be the equivalent of a single slide in a presentation. The file format is designed such that it is made up of collections

and parts. Each collection functions as a folder and each modular part functions as a file. These separate files are related together with relationships where each separate file is associated with a relationship type.

[0012] One embodiment is a file format for representing a presentation and/or a presentation template in a modular content framework. The modular content framework may include a file format container associated with the modular parts. The file format includes modular parts which are logically separate but associated with one another by one or more relationships. Each modular part is associated with a relationship type and the modular parts include a presentation part representing a start part for the presentation and/or the presentation template, a slide master part associated with the presentation part and/or the presentation template and operative to specify main components of the presentation, and a slide layout part associated with the slide master part and operative to specify a structure of the presentation. Each modular part is capable of being interrogated separately without other modular parts being interrogated, which offers gains in efficiency when the presentation is queried.

[0013] The modular parts may also include a document properties part containing built-in properties associated with the file format, a thumbnail part containing thumbnails associated with the file format, and a slide part containing a slide in the presentation. Each modular part is capable of being extracted from and/or copied from the presentation and reused in a different presentation along with associated modular parts identified by traversing the relationships of the modular part reused.

[0014] Another embodiment is a method for representing a presentation in a file format wherein modular parts associated with the presentation include each part written into the file format. The method involves writing a presentation part of the file format, querying the presentation for a slide master relationship type, and writing a slide master part of the file format separate from the presentation part. The method also involves establishing a relationship between the slide master part and the presentation part, writing a slide layout part separate from the slide master part, and establishing a relationship between the slide layout part and the slide master part.

[0015] Additionally, the method may involve establishing a relationship between the presentation part and a file format container where the file format container includes a document properties part containing built-in properties associated with the file format and a thumbnail part containing thumbnails associated with the file format. The method also involves writing a slide part containing a slide for the presentation separate from the slide master part and slide layout and establishing a relationship between the slide part and the slide layout.

[0016] Still further, the method may involve writing modular parts associated with relationship types wherein the modular parts that are to be shared are written only once and establishing relationships between the modular parts written. Writing the modular parts may also involve

examining each modular part, determining whether a change has occurred to the modular part examined, and when a change has occurred to the modular part examined, writing the modular part examined including the changes.

[0017] Still another embodiment is a computer program product including a computer-readable medium having control logic stored therein for causing a computer to represent a presentation in a file format where modular parts of the file format include each part written into the file format. The control logic includes computer-readable program code for causing the computer to write a presentation part of the file format, query the presentation for a slide master relationship type, write a slide master part of the file format separate from the presentation part, and establish a relationship between the slide master part and the presentation part. The control logic also includes computer-readable program code for causing the computer to write a slide layout part separate from the slide master part and establish a relationship between the slide layout part and the slide master part.

[0018] The computer program product also includes computer-readable program code for causing the computer to establish a relationship between the presentation part and a file format container. The file format container includes a document properties part containing built-in properties associated with the file format and a thumbnail part containing thumbnails associated with the file format. The computer-readable program code is also operative to cause the computer to write a slide part containing a slide for the presentation separate from the slide master part and slide layout part and establish a relationship between the slide part and the slide layout part.

[0019] The invention may be implemented utilizing a computer process, a computing system, or as an article of manufacture such as a computer program product or computer readable media. The computer program product may be a computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

[0020] These and various other features, as well as advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021]

FIGURE 1 is a computing system architecture illustrating a computing apparatus utilized in and provided by various illustrative embodiments of the invention;

FIGURES 2a-2b are block diagrams illustrating a

presentation relationship hierarchy for various modular parts utilized in a file format for representing presentations according to various illustrative embodiments of the invention;

FIGURES 3a-3b are tables illustrating relationship types and collection types utilized in the file format according to various illustrative embodiments of the invention; and

FIGURES 4-5 are illustrative routines performed in representing presentations in a modular content framework according to illustrative embodiments of the invention.

DETAILED DESCRIPTION

[0022] Referring now to the drawings, in which like numerals represent like elements, various aspects of the present invention will be described. In particular, FIGURE 1 and the corresponding discussion are intended to provide a brief, general description of a suitable computing environment in which embodiments of the invention may be implemented. While the invention will be described in the general context of program modules that execute in conjunction with program modules that run on an operating system on a personal computer, those skilled in the art will recognize that the invention may also be implemented in combination with other types of computer systems and program modules.

[0023] Generally, program modules include routines, programs, operations, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0024] Referring now to FIGURE 1, an illustrative computer architecture for a computer 2 utilized in an embodiment of the invention will be described. The computer architecture shown in FIGURE 1 illustrates a computing apparatus, such as a server, desktop, laptop, or handheld computing apparatus, including a central processing unit 5 ("CPU"), a system memory 7, including a random access memory 9 ("RAM") and a read-only memory ("ROM") 11, and a system bus 12 that couples the memory to the CPU 5. A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the ROM 11. The computer 2 further includes a mass storage device 14 for storing an operating system 16, application programs, and other pro-

gram modules, which will be described in greater detail below.

[0025] The mass storage device 14 is connected to the CPU 5 through a mass storage controller (not shown) connected to the bus 12. The mass storage device 14 and its associated computer-readable media provide non-volatile storage for the computer 2. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed by the computer 2.

[0026] By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, digital versatile disks ("DVJS"), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 2.

[0027] According to various embodiments of the invention, the computer 2 may operate in a networked environment using logical connections to remote computers through a network 18, such as the Internet. The computer 2 may connect to the network 18 through a network interface unit 20 connected to the bus 12. It should be appreciated that the network interface unit 20 may also be utilized to connect to other types of networks and remote computer systems. The computer 2 may also include an input/output controller 22 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 1). Similarly, an input/output controller 22 may provide output to a display screen, a printer, or other type of output device.

[0028] As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 14 and RAM 9 of the computer 2, including an operating system 16 suitable for controlling the operation of a networked personal computer, such as the WINDOWS XP operating system from MICROSOFT CORPORATION of Redmond, Washington. The mass storage device 14 and RAM 9 may also store one or more program modules. In particular, the mass storage device 14 and the RAM 9 may store a presentation application program 10. The presentation application program 10 is operative to provide functionality for the creation and layout of presentations and/or templates for presentations, such as a presentation 27, in an open file format 24, such as an XML file format. According to one embodiment of the invention, the presentation application program 10

and other application programs 26 comprise the OFFICE suite of application programs from MICROSOFT CORPORATION including the WORD, EXCEL, and POWERPOINT application programs.

[0029] Referring now to FIGURES 2a-2b, block diagrams illustrating a presentation relationship hierarchy 208 for various modular parts utilized in the file format 24 for representing a presentation and/or a presentation template according to various illustrative embodiments of the invention will be described. The presentation relationship hierarchy 208 lists specific presentation application relationships some with an explicit reference indicator 205 indicating an explicit reference to that relationship in the content of the modular part, for example via a relationship identifier. Non-explicit indicators 206, indicate that the modular part may potentially utilize features from the destination part without an explicit reference. Optional relationships with respect to validation are indicated in italics, and dashed connecting lines 203 indicate a one to many relationship. Thus, for example there is a slide part 222 for each slide associated with a presentation.

[0030] The various modular parts or components of the presentation hierarchy 208 are logically separate but are associated by one or more relationships. Each modular part is also associated with a relationship type and is capable of being interrogated separately without other modular parts being interrogated. A modular content framework may include a file format container 212 associated with the modular parts. The modular parts include, the presentation part 210 representing a start part for a presentation, a document properties part 214 containing built-in properties associated with the file format 24, and a thumbnail part 216 containing thumbnails associated with the file format 24.

[0031] The modular parts also include a slide master part 225 associated with the presentation part 210 and operative to specify main components of the presentation and a slide layout part 224 associated with the slide master part 225 and operative to specify a structure of the presentation. The slide layout part 224 may include a default layout associated with the presentation part 210. The presentation part 210 is associated with the slide master part in a one to many relationship meaning a slide master part 225 will exist for each slide grouping in the file format 24. Further, the slide master part 225 is associated with the slide layout part 224 in a one to many relationship meaning a slide layout part 224 will exist for each slide layout in the presentation. Other modular parts include a slide part 222 containing a slide in the presentation. The presentation part 210 is associated with the slide part 222 in a one to many relationship meaning a slide part 222 will exist for each slide in the presentation. It should be appreciated that each modular part is capable of being extracted and/or copied from the presentation and reused in a different presentation along with associated modular parts. Associated modular parts are identified when the presentation application traverses inbound and outbound relationships of the modular part

reused.

[0032] Still further, referring to FIGS. 2a-2b, other modular parts may include a style sheet part 227 containing data associated with a style of the presentation, a notes slide part 228 containing notes associated with the slide part 222, a mail envelope part 218 containing envelope data where a user of the presentation has sent the presentation via electronic mail, and a code file part 220, such as a visual basic application (VBA) project part containing VBA project code associated with the presentation. Other modular parts may include an image part 230 containing image data associated with the presentation, a sound part 234 containing audio data associated with the presentation, a movie part 232 containing movie data associated with the presentation, and an embedded object part 235 containing an object associated with the presentation. It should be appreciated that modular parts that are shared in more than one relationship are only written to memory once.

[0033] Other modular parts include a user data part 237 containing customized data capable of being read into the presentation and changed, a font part 244 containing one or more fonts associated with the presentation, a legacy drawing object part 238, such as an Escher 1.0 object and a drawing object part 242, such as an Escher 2.0 object both containing an object built using a drawing platform.

[0034] Still other modular parts may include a handout master part 221 containing data associated with defining a style of handouts associated with the presentation, a notes master part 229 containing data associated with defining a style of a note page associated with the notes, a comments part 245 containing comments associated with the presentation, and a thumbnail part 247 containing thumbnails associated with the presentation. It should be appreciated that certain modular parts are global and thus, can be used anywhere in the file format. In contrast, some modular parts are non-global and thus, can only be shared on a limited basis.

[0035] In various embodiments of the invention, the file format 24 may be formatted according to extensible markup language ("XML") and/or a binary format. As is understood by those skilled in the art, XML is a standard format for communicating data. In the XML data format, a schema is used to provide XML data with a set of grammatical and data type rules governing the types and structure of data that may be communicated. The XML data format is well-known to those skilled in the art, and therefore not discussed in further detail herein.

[0036] FIGURES 3a-3b are tables 300 illustrating relationship types and table 350 illustrating collection types utilized in the file format 24 according to various illustrative embodiments of the invention. A relationship type table 300 lists the relationship types associated with the modular parts. The relationship types not only identify an association or dependency but also identify the basis of the dependency. The relationship types include a font relationship 302, a code file relationship 303 capable of

identifying potentially harmful code files, and a user data relationship 304. The relationship types also include a style sheet relationship 307, a comments relationship 308, an embedded object relationship 310, a legacy drawing object relationship 314, a drawing object relationship 312, an image relationship 317, a sound relationship 318, and a movie relationship 320. Still further, the relationship types may include a mail envelope relationship 322, a document properties relationship 324, a thumbnail relationship 325, a slide relationship 327, a notes slide relationship 328, a slide master relationship 330, and a slide layout relationship 332. Other relationship types include a handout master relationship 335, a notes master relationship 337, and a presentation relationship 340.

[0037] Referring to FIGURES 2a-2b and 3b, the table 350 lists collection types for organizing the modular parts. The collection types include a drawings collection 602 that includes the drawing object parts 238 and 242, a fonts collection 604 that includes the font part 244, an images collection 605 that includes the image part 230, and a slide layouts collection 607 that includes the slide layout part 224. The collection types also include a masters collection 610 that includes the slide master part 225, a media collection 612 that includes the movie and the sound parts 232 and 234, and a notes slides collection 614 that collection includes the note slides part 228.

[0038] Still further, the collection types may include a slides collection 617 that includes the slide part 222, a styles collection 619 that includes the styles sheet part 227, an embeddings collection 620 that includes the embedded object part 235 and the user data part 237, a comments collection 622 including the comments part 245, a subdocuments collection 624, a handout masters collection 625, and a notes masters collection 627.

[0039] FIGURES 4-5 are illustrative routines performed in representing presentations in a modular content framework according to illustrative embodiments of the invention. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments of the present invention are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated in FIGURES 4-5, and making up the embodiments of the present invention described herein are referred to variously as operations, structural devices, acts or modules. It will be recognized by one skilled in the art that these operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims set forth herein.

[0040] Referring now to FIGURES 2a, 2b and 4, the routine 700 begins at operation 704, where the presentation application program 10 writes the presentation part 210. The routine 700 continues from operation 704 to operation 705, where the presentation application program 10 queries the presentation for slide master relationships. Next, at operation 707, the presentation application writes the slide master parts 225 referenced in the presentation part 210 and establishes relationships between each slide master part 225 and the presentation part 210.

[0041] Next, at operation 708, the presentation application 10 writes slide layout parts for each slide master part 225 and establishes relationships between the slide layout parts and the slide master parts. Then at operation 710, the presentation application 10 writes a style sheet part for each slide master and establishes a relationship between the slide master part and the style sheet part. The routine 700 then continues to operation 712, where the presentation application writes all slide parts for the presentation and establishes a relationship between the slide parts and the corresponding slide layout part.

[0042] Then at operation 714, the presentation application 10 writes other modular parts associated with relationship types, such as the sound part, the image part, and the movie part associated with previously written modular parts. Any modular part to be shared between other modular parts is written only once. The routine 700 then continues to operation 722.

[0043] At operation 722, the presentation application establishes relationships between newly written and previously written modular parts. The routine 700 then terminates at return operation 727.

[0044] Referring now to FIGURE 5, the routine 800 for writing modular parts will be described. The routine 800 begins at operation 802 where the presentation application 10 examines data in the presentation application. The routine 800 then continues to detect operation 804 where a determination is made as to whether the data has been written to a modular part. When the data has not been written to a modular part, the routine 800 continues from detect operation 804 to operation 805 where the presentation application writes a modular part including the data examined. The routine 800 then continues to detect operation 807 described below.

[0045] When at detect operation 804, the data examined has been written to a modular part, the routine 800 continues from detect operation 804 to detect operation 807. At detect operation 807 a determination is made as to whether all the data has been examined. If all the data has been examined, the routine 800 returns control to other operations at return operation 812. When there is still more data to examine, the routine 800 continues from detect operation 807 to operation 810 where the presentation application 10 points to other data. The routine 800 then returns to operation 802 described above.

[0046] Based on the foregoing, it should be appreciated that the various embodiments of the invention include

file formats, methods and computer program products for representing presentations in a modular content framework. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

Claims

1. A file format for representing at least one of a presentation and a presentation template, the file format representing at least one of the presentation and the presentation template in a modular content framework implemented within a computing apparatus, the file format comprising:

modular parts logically separate but associated by one or more relationships wherein each modular part is associated with a relationship type and wherein the modular parts include:

a presentation part representing a start part for at least one of the presentation and the presentation template; and
a slide master part associated with the presentation part and operative to specify main components of at least one of the presentation and the presentation template;

wherein each modular part is capable of being interrogated separately without other modular parts being interrogated.

2. The file format of claim 1, wherein the modular content framework includes a file format container associated with the modular parts wherein the modular parts further include:

a document properties part containing built-in properties associated with the file format; and
a thumbnail part containing thumbnails associated with the file format.

3. The file format of claim 1, wherein the modular parts further include at least one of the following:

a slide layout part associated with the slide master part and operative to specify a structure of the presentation; and
a slide part containing a slide in the presentation;

wherein a modular part along with associated modular parts identified by traversing relationships of the modular part are capable of being one of extracted from and copied from the presentation and reused

in a different presentation.

4. The file format of claim 3, wherein the modular parts further include at least one of the following:

a style sheet part containing data associated with a style of a referring modular part;
a handout master part containing data associated with defining a style of handouts associated with the presentation;
a notes slide part containing notes associated with the slide;
a notes master part containing data associated with defining a style of a note slide associated with the notes;
a comments part containing comments associated with the presentation;
a thumbnail part containing thumbnails associated with the presentation;
a mail envelope part containing envelope data where a user of the presentation has sent the presentation via electronic mail; and
a code file part containing code associated with the presentation.

5. The file format of claim 4, wherein the modular parts further include at least one of the following:

an image part containing image data associated with the presentation;
a sound part containing audio data associated with the presentation;
a movie part containing movie data associated with the presentation;
an embedded object part containing an object associated with the presentation;
a user data part containing customized data capable of being read into the presentation and changed;
a font part containing data that defines fonts associated with the presentation; and
a drawing object part containing an object built using a drawing platform.

6. The file format of claim 5, wherein at least some of the modular parts are organized in collection types and wherein the collection types include at least one of the following:

a comments collection wherein the comments collection includes the comments part;
a drawings collection wherein the drawings collection includes the drawing object part;
a fonts collection wherein the fonts collection includes the font part;
a subdocument collection wherein the subdocument collection includes a modular parts that contain embedded documents associated with

- other modular parts in the presentation;
 an images collection wherein the images collection includes the image part;
 a slide layouts collection wherein the slide layouts collection includes the slide layout part;
 a handout masters collection wherein the handout masters collection includes the handout master part;
 a slide masters collection wherein the masters collection includes the slide master part;
 a notes masters collection wherein the masters collection includes the notes master part;
 a media collection wherein the media collection includes the movie and the sound parts;
 a notes slides collection wherein the notes slides collection includes the note slides part;
 a slides collection wherein the slides collection includes the slide part;
 a styles collection wherein the styles collection includes the styles sheet part; and
 an embeddings collection wherein the embeddings collection includes the embedded object part and the user data part.
7. The file format of claim 3, where the relationship types associated with the modular parts comprise at least one of a font relationship, a code file relationship capable of identifying potentially harmful code files, a user data relationship, a style sheet relationship, a comments relationship, an embedded object relationship, a drawing object relationship, an image relationship, a sound relationship, a movie relationship, a mail envelope relationship, a document properties relationship, a thumbnail relationship, a slide relationship, a notes slide relationship, a slide master relationship, a slide layout relationship, and a file format container embedded object relationship.
8. The file format of claim 3, wherein content of the slide is capable of being one of extracted from and copied from the presentation and reused in a different presentation.
9. The file format of claim 3, wherein each modular part is capable of being locked separately while the other modular parts remain available for locking whereby multiple editors may each concurrently edit a modular part of the file format.
10. The file format of claim 3, wherein the modular parts are capable of providing semantics about content within the presentation when a modular part is interrogated.
11. The file format of claim 3, wherein the slide is authored in an authoring context of the presentation and wherein the slide is capable of being one of extracted from and copied from the presentation and
- moved to a different presentation and wherein the slide is further capable of being altered to appear as though the slide was authored in an authoring context of the different presentation.
12. The file format of claim 4, wherein the file format is capable of providing a high-resolution thumbnail preview of each slide in the presentation.
13. The file format of claim 1, wherein the file format is formatted according to at least one of a markup language format and a binary format.
14. A method for representing a presentation in a file format wherein modular parts associated with the presentation include each part written into the file format, the method comprising:
 writing a presentation part of the file format;
 querying the presentation part for a slide master relationship type;
 writing a slide master part of the file format separate from the presentation part;
 establishing a relationship between the slide master part and the presentation part;
 writing a slide layout part separate from the slide master part; and
 establishing a relationship between the slide layout part and the slide master part.
15. The method of claim 14, further comprising establishing a relationship between the presentation part and a file format container wherein the file format container includes:
 a document properties part containing built-in properties associated with the file format; and
 a thumbnail part containing thumbnails associated with the file format.
16. The method of claim 14, further comprising writing a slide part separate from the slide master part and containing a slide for the presentation and establishing a relationship between the slide part and the slide layout part.
17. The method of claim 16, further comprising:
 writing modular parts associated with relationship types wherein the modular parts that are to be shared are written only once; and
 establishing relationships to the modular parts written.
18. The method of claim 17, wherein writing the modular parts associated with the relationship types comprises at least one of the following:

writing a style sheet part containing data associated with a style of the presentation and establishing a relationship between the style sheet part and the slide master part;

writing a notes slide part containing notes associated with the slide and establishing a relationship between the notes slide part and the slide part; 5

writing a mail envelope part containing electronic mail envelope data and establishing a relationship between the mail envelope part and the presentation part; and 10

writing a code file part containing code associated with the presentation and establishing a relationship between the code file part and the presentation part. 15

19. The method of claim 17, wherein writing the modular parts associated with the relationship types comprises: 20

a) examining data associated with the presentation;

b) determining whether the data examined has been written to a modular part; 25

c) when the data examined has not been written to the modular part, writing the modular part to include the data examined, examining other data associated with the presentation, and repeating b) through d); and 30

d) when the data examined has been written to the modular part, examining other data and repeating b) through d).

20. A computer program product comprising a computer-readable medium having control logic stored therein for causing a computer to represent a presentation in a file format wherein modular parts of the file format include each part written into the file format, the control logic comprising computer-readable program code for causing the computer to: 35 40

write a presentation part of the file format;

query the presentation part for a slide master relationship type; 45

write a slide master part of the file format separate from the presentation part;

establish a relationship between the slide master part and the presentation part;

write a slide layout part separate from the slide master part; and 50

establish a relationship between the slide layout part and the slide master part.

55

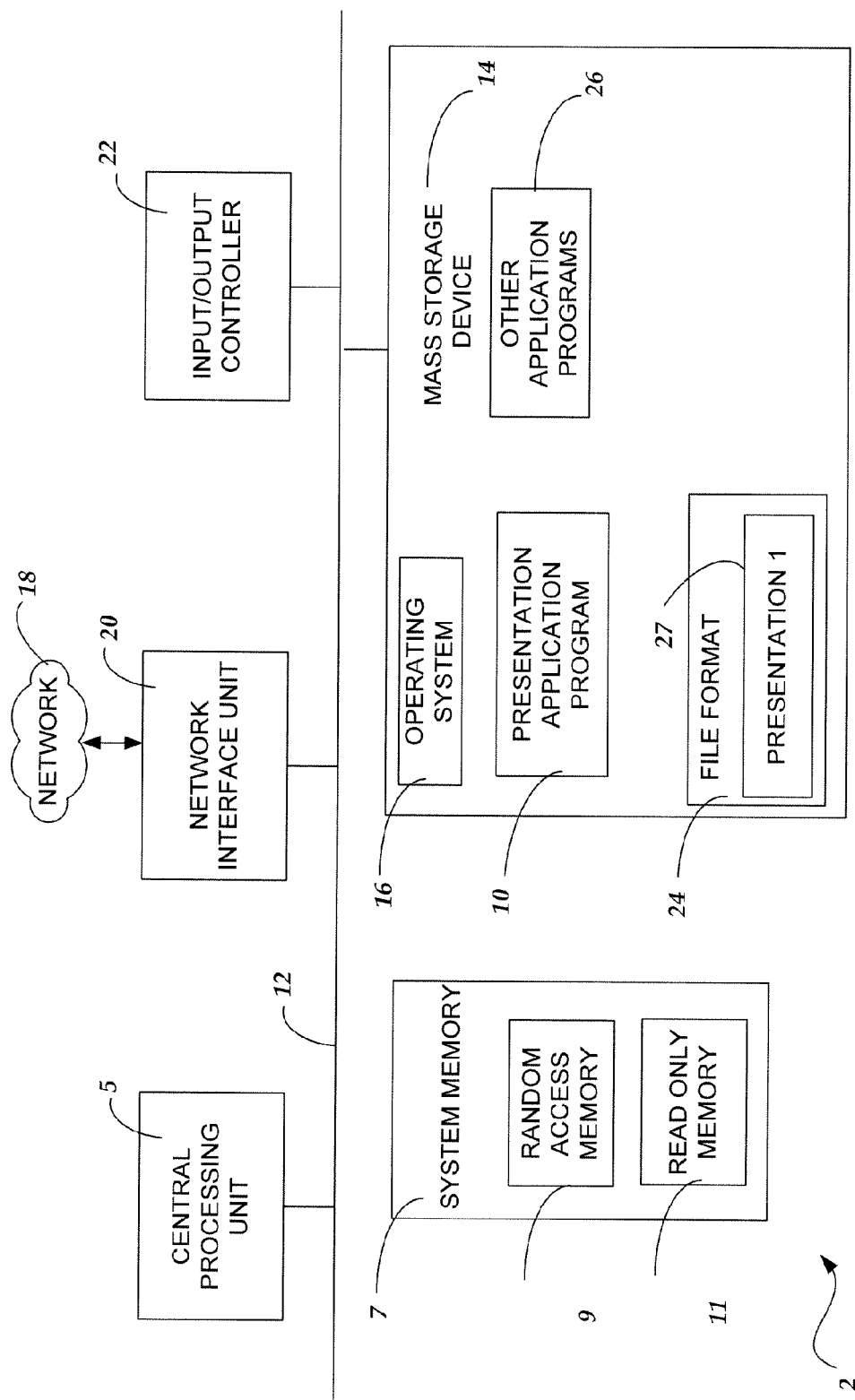


Fig. 1

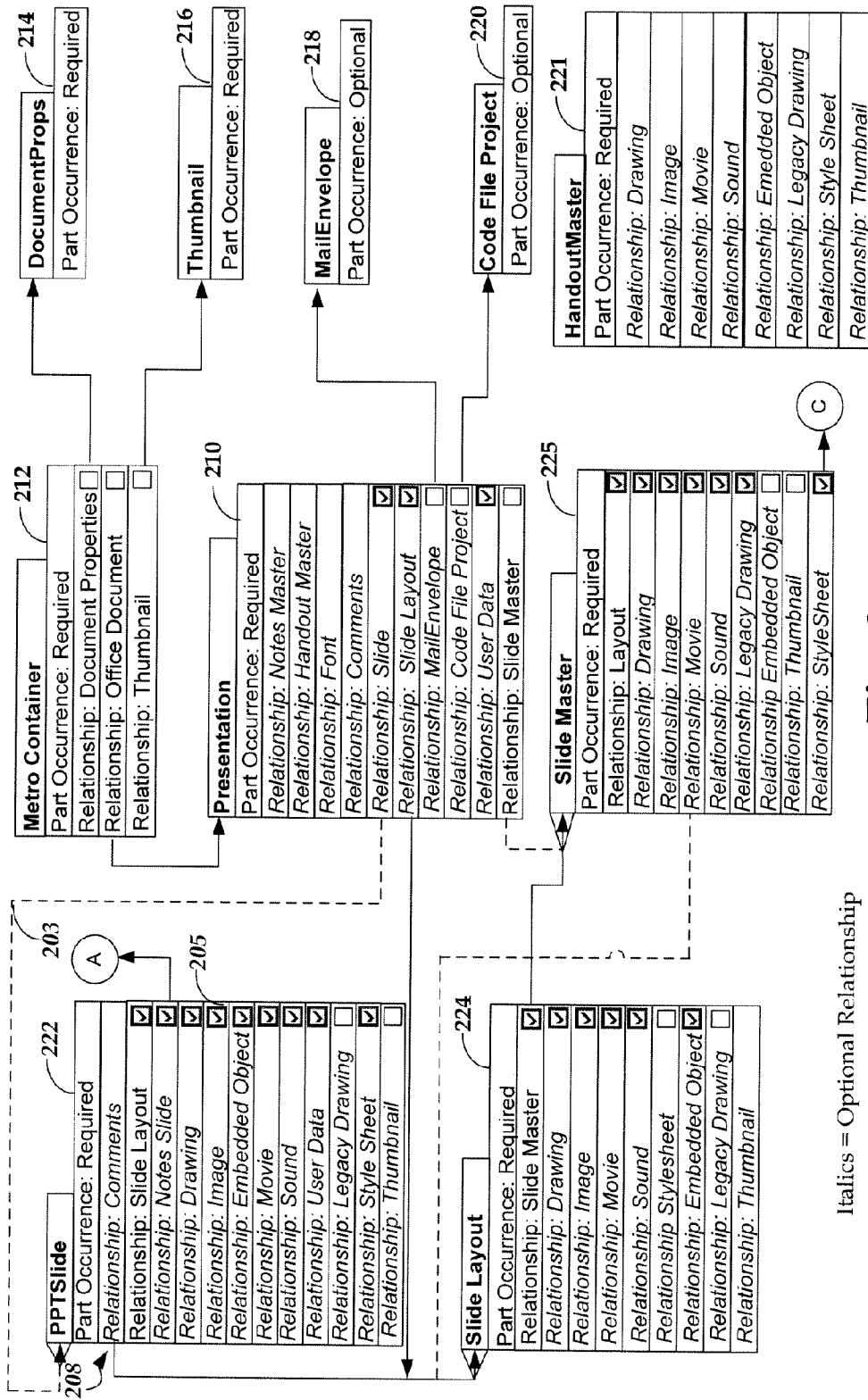


Fig. 2a

Italics = Optional Relationship
 - - - = 1 to N Relationship

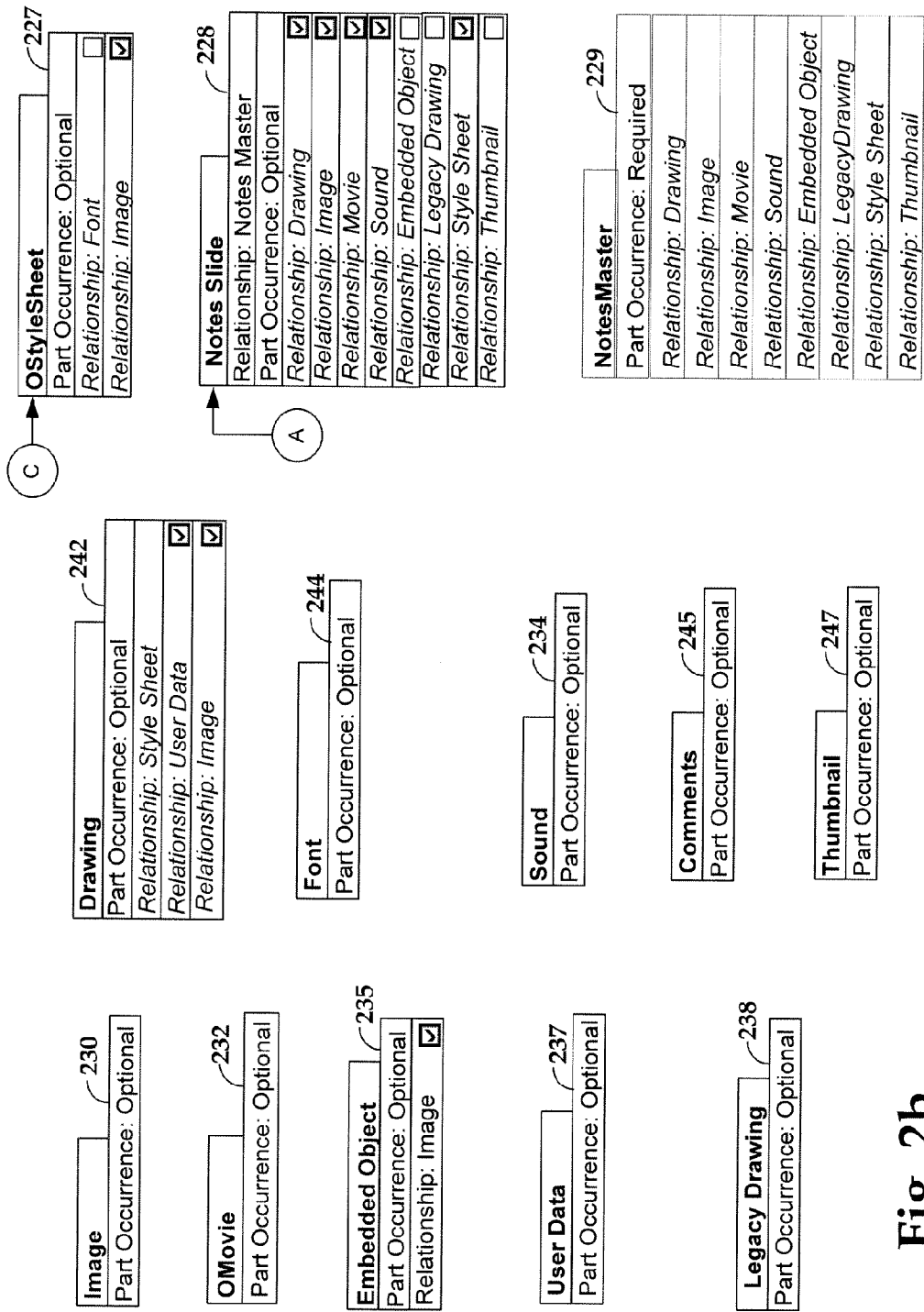


Fig. 2b

300

Relationship Types Table

Relationship	Type/Comments
codeFile	The target of this relationship is a part that contains code such as Visual Basic for Applications (VBA) projects.
comments	The target of this relationship is a part that contains the comments for the referring part.
docProps	The target of this relationship is a part that contains all of the built-in document properties for the container.
drawing	The target of this relationship is a part that contains a drawing found on the referring part.
font	The target of this relationship is a part that contains the definition of a font in use by the referring part.
handoutMaster	The target of this relationship is a part that contains the handout master which defines the initial look and feel of handouts for the referring presentation.
image	The target of this relationship is a part that contains an image (vector/raster) on the referring part.
legacyDrawing	The target of this relationship is a part that contains a drawing found on the referring part.
mailEnvelope	The target of this relationship is a part that contains mail envelope information (who the file was sent to, what the subject line was, etc.) for the referring presentation.
movie	The target of this relationship is a part that contains a movie found on the referring part.
notesMaster	The target of this relationship is a part that contains the notes master which defines the initial look and feel of notes pages for the referring part.
notesSlide	The target of this relationship is a part that contains the notes slide for the referring part.
embeddedObject	The target of this relationship is a part that contains an embedded document found on the referring part.
pptDocument	The target of this relationship is a part that contains the primary presentation part for a PowerPoint presentation.
slide	The target of this relationship is a part that contains a slide associated with the referring part.
slideLayout	The target of this relationship is a part that contains a slide layout which defines the initial look and feel for a slide part.
slideMaster	The target of this relationship is a part that contains the slide master which defines the initial look and feel of a slide or a slide layout.
sound	The target of this relationship is a part that contains a sound found on the referring part.
styleSheet	The target of this relationship is a part that contains a style sheet that defines basic formatting characteristics available to the referring part.
thumbnail	The target of this relationship is a part that contains a thumbnail image for the container.
userData	The target of this relationship is a part that contains a block of user-defined XML available for use by the referring container.

Fig 3a

350

Collection Types Table

Token	Comments
622 comments	This collection holds the parts that contain comments associated with other parts in the presentation (e.g. the presentation part, slide parts, etc.).
602 drawings	This collection holds any parts that contain simple drawings associated with other parts in the presentation (e.g. the various master and slide parts, etc.).
620 embeddings	This collection holds any parts that contain general-purpose blocks of data to be used by other parts in the presentation.
604 fonts	This collection holds any parts that contain definitions for certain fonts used in other parts of the presentation.
625 handoutMasters	This collection holds all of the handoutMaster parts for the presentation.
605 images	This collection holds all of the image parts associated with other parts in the presentation (e.g. the various master parts, slide parts, etc.).
607 slideLayouts	This collection holds all of the slide layout parts for all of the slide masters in the presentation.
610 slideMasters	This collection holds all of the slide master parts for a presentation.
624 subDocuments	This collection holds all of the parts that contain embedded documents associated with other parts in the presentation (e.g., the various master and slide parts, etc.).
612 media	This collection holds all of the movie and sound parts associated with other parts in the presentation (e.g., the various master and slide parts, etc.).
627 notesMasters	This collection holds all of the notesMaster parts used by or available to the notesSlides parts.
614 notesSlides	This collection holds all of the notesSlide parts for the slide parts in the presentation.
617 slides	This collection holds all of the slide parts for all of the slides in the presentation.
619 styles	This collection holds all of the style parts available to this presentation.

Fig 3b

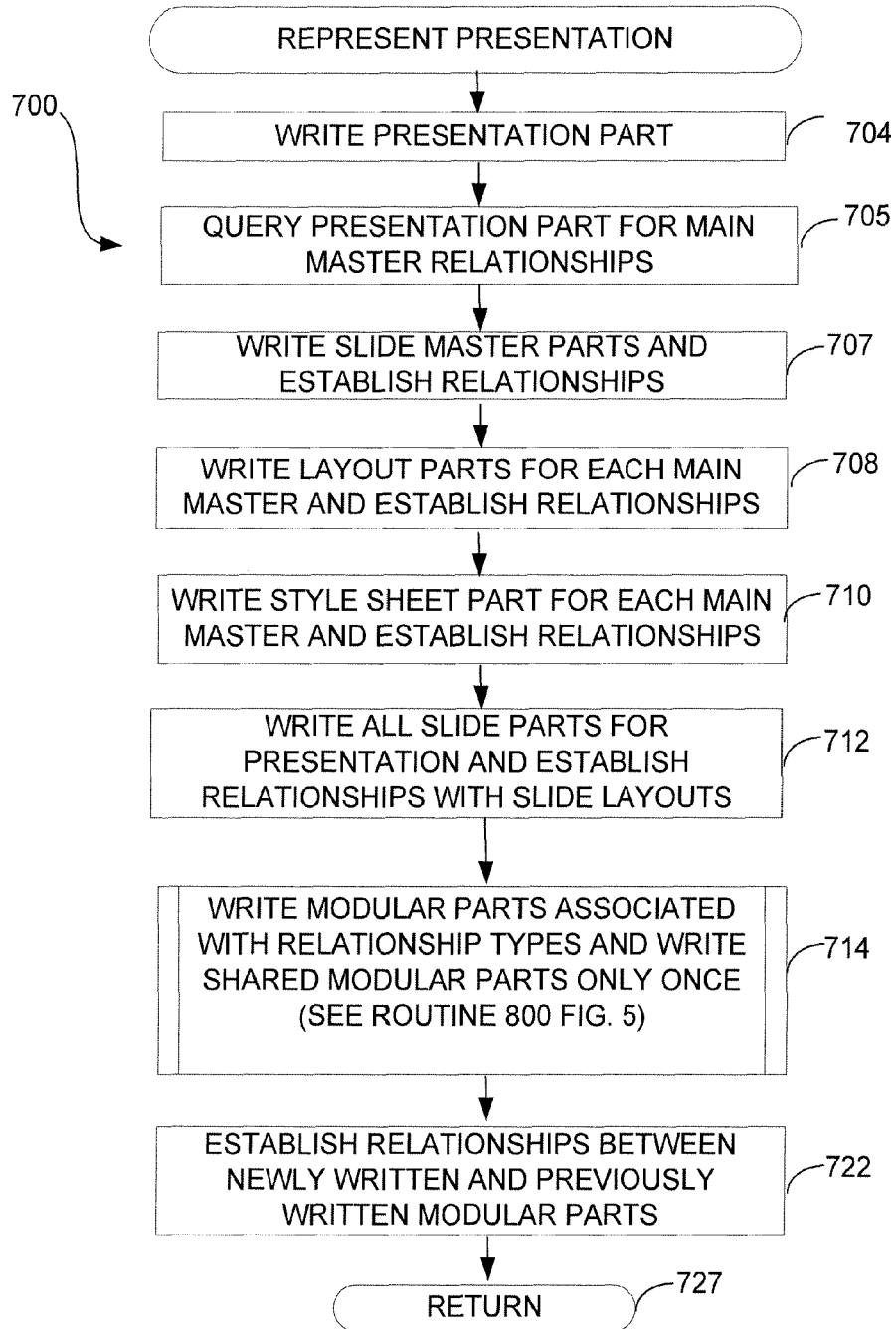


Fig. 4

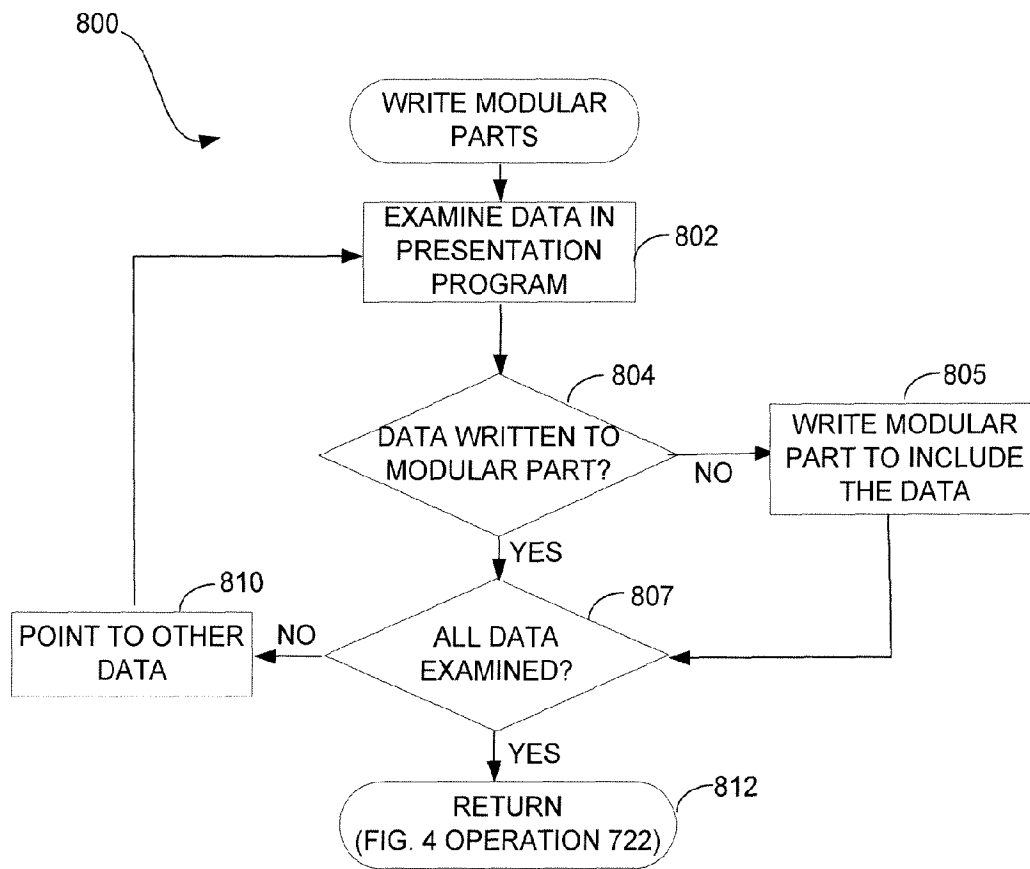


Fig. 5